



## UPPER CANADA TECHNOLOGIES



**IOPORT OCX version 1.1 Copyright © 1998, Robert Woudsma  
All Rights Reserved**

### Why would I use IOPORT?

In DOS based variations of the BASIC programming language programmers could easily access the IBM-compatible computers I/O ports with the INP and OUT commands. For many programmers Visual Basic has become programming language of choice to develop Windows based applications. As robust and simple as Visual Basic is to use, it does not contain the commands to access the computers I/O ports. IOPORT is a custom control that can be easily added to any Visual Basic project to provide 8-bit or 16-bit READ and WRITE capability to any port within the PCs I/O address space. IOPORT is a 32-bit ActiveX /OCX control designed to be used with Visual Basic 4.0 or 5.0 under Windows 95. It will not operate properly under Windows NT.

### Shareware Notice

The IOPORT is distributed under shareware principle. As such, you are welcome to use it for a trial period of up to 30 days. If you find that IOPORT is useful to you and you wish to continue using it then please register this product and support software being produced under the shareware concept. You may freely copy/distribute the shareware version in it's original form as long as you make no changes to any of the IOPORT files and you charge no fee for it beyond reproduction costs. If you register IOPORT, then you may use it in your own personal software or in commercial products with no royalty fee.

### IOPORT.OCX Installation & Registration

1. If you downloaded IOPORT from the Internet copy the IOPORT.ZIP file to a temporary directory on your hard drive and unzip it. If you received IOPORT on a distribution diskette, then the files are not zipped and you can copy them directly from the diskette.

2. Copy the file **IOPORT.OCX** to your Windows\System or Win95\System directory. If the program **regsvr32.exe** is not in the System directory then copy this file also from the temporary directory or diskette to the System directory.
3. Next, you must register **IOPORT** with Windows 95. To do this, select the **Start** button and then choose **Run**. Type **regsvr32 ioport.ocx** to register the **IOPORT OCX**. A dialog box will appear indicating whether or not the registration was successful.
4. If you copied **IOPORT.OCX** to a different directory than the Windows\System directory you must include the full path when registering the control. For example, if you copied **IOPORT.OCX** to the directory **C:\program files\vb5\CONTROLS** then you must type the following command to register the control:

**regsvr32 c:\program files\vb5\controls\ioport.ocx**

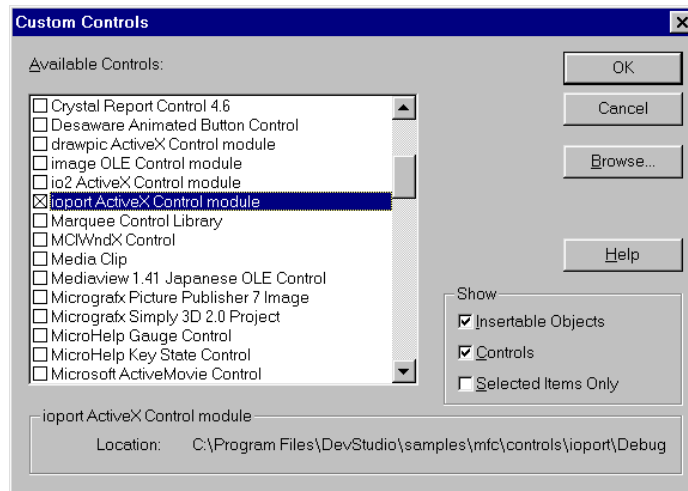
The regsvr32.exe program must still exist or be placed in the Windows\System or Win95\System directory.

5. A number of .DLL files must be present and registered in the Windows\System or Win95\System directory to be called when registering and using the **IOPORT OCX**. These files may or may not already exist in the directory. The files include: **MFC42.dll**, **MSVCRT40.dll** and **OLEPRO32.dll**. If the **IOPORT OCX** fails to become successfully registered using the methods outlined in sections 3 and 4 above. Download the ZIP file **sprtdlls.zip** from the UCT Web site or copy the ZIP file from the distribution diskette. Unzip this file to the Windows\System or Win95\System directory. Use the **regsvr32.exe** program as outlined in section 3 above to register **OLEPRO32.dll** and **MFC42.dll** and repeat the registration process for **IOPORT.OCX**.

## Adding IOPORT to your Visual Basic Project

### 1. Visual Basic 4.0

Once the IOPORT OCX has been registered with Windows 95, adding it to the Visual Basic controls toolbox is simple. From the Visual Basic main menu select **Tools**. From the **Tools** drop-down menu select **Custom Controls...** This will display the Custom Controls dialog box, as shown below which lists the controls that can be

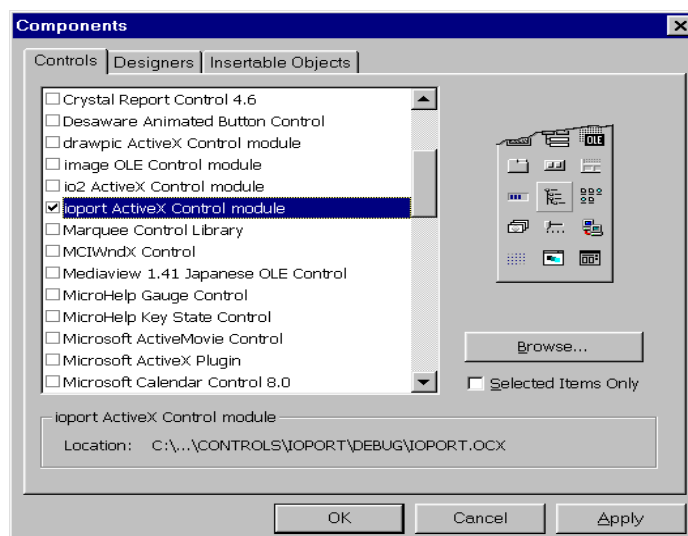


added to the Visual Basic toolbox. Select **ioport ActiveX Control module** from the list and click on OK. The dialog box will disappear. You should now see a new control icon, as shown below, in the controls toolbox.



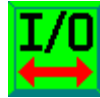
### 2. Visual Basic 5.0

Accessing the IOPORT control is a bit different in Visual Basic 5.0. Instead of selecting **Tools** from the main menu, select **Project**. From the Project drop-down menu select **Components**. A similar Components dialog box will appear as shown below.



On the **Controls** page of the dialog box select the **ioport ActiveX Control module** and click on OK. The control icon above should now be displayed in the toolbox.

To place the IOPORT control on a form follow the procedure for most Visual Basic controls; click on the control icon in the toolbox and then use the mouse to drag a rectangle on the form. When the rectangle is drawn, the control will be displayed as shown below. The properties of the IOPORT control are now available to your Visual Basic code.

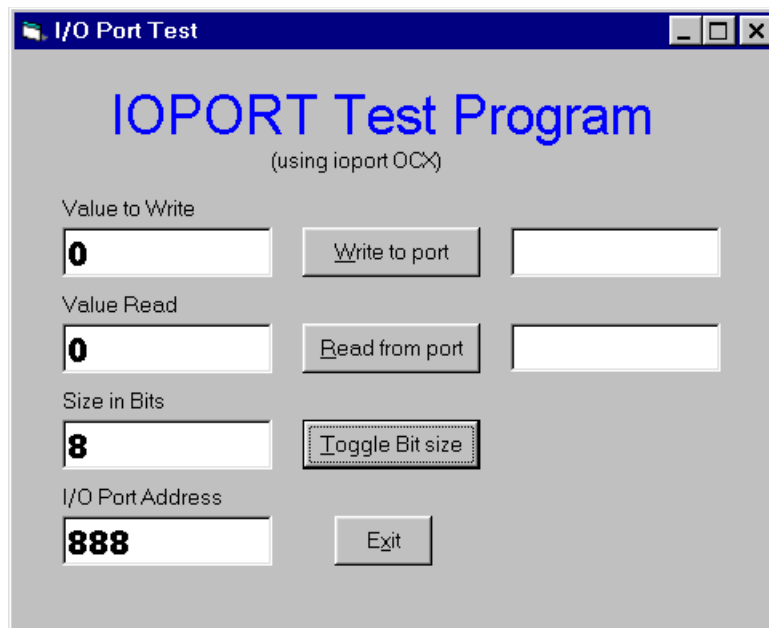


NOTE: The control is visible only at design time on your Visual Basic form. It will disappear from the form when the Visual Basic program is executed.

### Using IOPORT in Visual Basic

#### 1) IOTEST Sample Application

There are two Visual Basic projects included in the IOPORT.ZIP file or on the distribution disk; one called IOTEST5 for use with Visual Basic 5.0 and another called IOTEST4 for use with Visual Basic 4.0. The test application form appears as is shown below.



The test application files include the form, project and executable files which are thoroughly documented. By viewing and running these files you can get a understanding of how the IOPORT OCX can be used in your own Visual Basic applications.

## 2) IOPORT Properties

The NAME property of the IOPORT.OCX control is IOPORT. When the control is first placed on a Visual Basic form it's default NAME property will be set to IOPORT1, following the control naming convention of Visual Basic. If a second IOPORT control is placed on the form it's NAME property would be set to IOPORT2. These names can be changed in the IOPORT property window but this should be done before any code is written using the IOPORT's properties.

Besides the general properties of almost all Visual Basic controls such as NAME, TOP, LEFT, ABOUT, INDEX etc. there are four unique properties in the IOPORT control; **address**, **bitsize**, **USValue** and **value**. A description of each control property and how it is used follows.

- **Address** This property sets the address of the I/O port that the programmer wishes to read from or write to. The value can be an integer number from 0 to 65535.
- **Bitsize** This property defines whether the value to be written or read will be a byte (8-bits) or a word (16-bits). The only valid values for this property is 8 or 16, any other values will result in no change of the property.
- **Value**
  - 1) 8-bit operation: (bitsize = 8)**  
During a write operation the value contained in the **value** property will be delivered to the I/O port defined by the **address** property. Valid write values are from 0 to 255.  
During a read operation the value of the contents of the I/O port defined by the **address** property will be returned to the **value** property. Values returned will be from 0 to 255.
  - 2) 16-bit operation: (bitsize = 16)**  
In 16-bit operation the **value** property operates the same as in 8-bit mode. The only difference is that in 16-bit mode the values written or read are considered to be signed 16-bit words. As such valid write and returned values range from -32768 to +32767.
- **USValue** **16-bit mode only: (bitsize = 16)**  
This property operates in the same way as the value property, above but uses unsigned 16-bit values and is only valid in 16-bit mode. When in 16-bit mode it may be desired by programmers that the 16-bit value being written or returned be of the unsigned integer format where valid values range from 0 to 65535. The form below shows the IOTEST in 16-bit operation.

I/O Port Test

# IOPORT Test Program

(using ioport OCX)

Signed value Write	<input type="text"/>	Unsigned value Write	<input type="text"/>
<input type="button" value="Write to port"/>			
Signed value Read	<input type="text" value="-22360"/>	Unsigned value Read	<input type="text" value="43176"/>
<input type="button" value="Read from port"/>			
Size in Bits	<input type="text" value="16"/>	<input type="button" value="Toggle Bit size"/>	
I/O Port Address	<input type="text" value="45"/>	<input type="button" value="Exit"/>	

## **IOPORT Software Registration**

Registration of the IOPORT control with Upper Canada Technologies grants you a single user license. As such, you may install IOPORT on as many computers as you wish provided that only the registered person uses it and that no two copies of IOPORT will be in use at the same time. You are required to purchase one copy of IOPORT for each person that will use it.

As a registered user you may include IOPORT OCX in any software that you develop and distribute so long as IOPORT is only a component of your software and does not constitute your software's major function.

To register, please fill out the registration form on the last page of this document and mail it, along with your payment, to:

Robert Woudsma  
UPPER CANADA TECHNOLOGIES  
R.R. #1, 2699 Lakefield Dr.  
Inverary, Ontario  
K0H 1X0  
E-mail: [rjw@uct.on.ca](mailto:rjw@uct.on.ca)

Registered users receive free updates to the current version (ie 1.x to 1.y). Major revision upgrades (ie 1.x to 2.x) will only cost \$5.00.

Constructive comments/criticism are always welcome at the above snail-mail or E-mail address.

## **IOPORT Pricing**

Copies	Price (includes Taxes)
1 to 9	\$15.00 each
10 to 19	\$13.50 each
20 to 29	\$12.00 each
30 to 50	\$10.50 each
50+	\$ 8.50 each

Site License arrangements are available on a per unit cost basis. Please E-mail at the above address for Site License pricing.

## **Educational Prices**

As an educator, I have a soft spot for accredited educational institutions, all of which seem to be undergoing fiscal restraint. Special pricing is in place for educational facilities on unit or site license basis. Please E-mail for details.

## Files Listing

The IOPORT distribution diskette and/or IOPORT.ZIP file should contain the following files:

IOPORT.OCX	the ActiveX control file
REGSVR32.EXE	Windows registration file for OCX controls
IOPORT.DOC	MS-WORD 7.0/Office 97 documentation file
IOPORT.TXT	DOS text documentation file

Visual Basic 4.0 IOTEST files:

IOTEST4.EXE	Executable IOTEST program
IOTEST4.FRM	Visual Basic form file
IOTEST4.VBP	Visual Basic project file
IOTEST4.VBW	Visual Basic Workspace file

Visual Basic 5.0 IOTEST files:

IOTEST5.EXE	Executable IOTEST program
IOTEST5.FRM	Visual Basic form file
IOTEST5.VBP	Visual Basic project file
IOTEST5.VBW	Visual Basic Workspace file

In addition, the distribution diskette contains the SPRTDLLS.ZIP file which contains the support .DLL type files required in the Windows\System or Win95\System directory to register and operate various OCX-type files including IOPORT.OCX.

This .ZIP file can also be downloaded from the UPPER CANADA TECHNOLOGIES web site at:

**[www.uct.on.ca](http://www.uct.on.ca)**



## IOPORT Registration Form

Please fill out the following form and mail with your payment to:

UPPER CANADA TECHNOLOGIES  
R.R.#1, 2699 Lakefield Dr.  
Inverary, Ontario Canada  
K0H 1X0

Name: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

E-Mail: \_\_\_\_\_

Copies	Price (includes taxes)	Number of copies: _____
1 to 9	\$15 each	Price per copy: _____
10 to 19	\$13.50 each	
20 to 29	\$12.00 each	Total price: _____
30 to 50	\$10.50 each	
50+	\$ 8.50 each	Amount enclosed: _____

How did you hear about IOPORT? \_\_\_\_\_

Where did you get IOPORT? \_\_\_\_\_

What will you use IOPORT for? \_\_\_\_\_

\_\_\_\_\_

Features you would like to see added: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Comments: \_\_\_\_\_

\_\_\_\_\_

Thank You for purchasing IOPORT.OCX !

## **IOPORT Revision History**

<b><u>Version</u></b>	<b><u>Date</u></b>	<b><u>Description</u></b>
1.0	02/98	IOPORT.OCX first released
1.1	08/98	Fixed bug in 8-bit data read section. Would read 8-bit data as 0 through 127, then -128 through -1.